Software Development Principles

Lecture 2
Python I/O

Lecturer:

Karen Nolan

karen.nolan@it-tallaght.ie

Python I/O - Topics

- .txt file
  - **Read** .txt files using basic open function
  - **Write** and **append** .txt files using basic open function
  - **Splitting data** into lists for future processing
- Using the ***with*** statement
- CSV file
  - **Opening** CSV files using the Python CSV import and ***with***
  - **Reading** CSV files
  - **Writing** to CSV files

Python I/O - .txt Files

- To start we need a .txt file!
- You have two choices for location:
  - Same folder as the python project
  - Any location but this (path) must be specified

Python I/O - .txt Files

- To start we need a .txt file!
- You have two choices for location:
  - Same folder as the python project

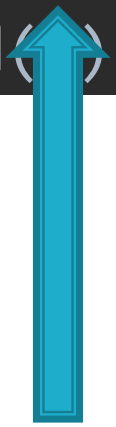
```
myData = open("myfile.txt", "r")
print(myData.read())
```

  - Any location but this (path) must be specified

```
myData = open("c://myfile.txt", "r")
print(myData.read())
```

Python I/O - .txt Files

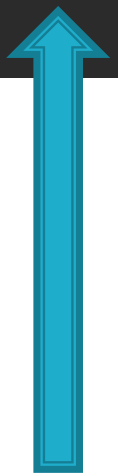
```
myData = open("myfile.txt", "r")
print(myData.read())
```



File Location

Python I/O - .txt Files

```
myData = open("myfile.txt", "r")
print(myData.read())
```



Open Action => r = Read

Python I/O - .txt Files

```
myData = open("myfile.txt", "r")
print(myData.read())
```

R	Read mode	File is only being read (write actions not allowed)
w	Write mode	Erases current file and rewrites as a new file
a	Appending mode	Adds new data to the end of the file
r+	Special read and write mode, which is used to handle both actions when working with a file, appends to the end of the file	

Python I/O - .txt Files

```
myData = open("myfile.txt", "r")
print(myData.read())

myData.close()
```

- Good practice to close the connection
- If left open errors may occur!

Python I/O – Class Example 1

- Download and open the file: **aCityOfTwoTails.txt**

by Charles Dickens

- Read and print the file
- Count the spaces
- Count the vowels

Python I/O – Solution

```
myData = open("aCityOfTwoTails.txt", "r")

text = myData.read()
myData.close()

print(text)

spaces = 0
vowels = 0
vowelsList = ["a", "e", "i", "o", "u", "A", "E", "I", "O", "U"]

for character in text:
    if character == " ":
        spaces += 1
    if vowelsList.__contains__(character):
        vowels += 1

print("----------------------------------------")
print("The number of spaces: ", spaces)
print("The number of vowels: ", vowels)
```

Python I/O – Class Example 2

- Download and open the file: **aCityOfTwoTails.txt**

by Charles Dickens

- Read and print the file
- Count the spaces
- Count the vowels
- **Create log file with spaces and vowels saved**

Python I/O – Solution

```
# Create Log
logFile = open("ACOTT_LogFile.txt", "w")

logFile.write("----------------------------------------")
logFile.write("\nThe number of spaces: " + str(spaces))
logFile.write("\nThe number of vowels: " + str(vowels))

logFile.close()
```

Python I/O – Class Example 3

- Now that we have seen a book, what about automated downloading of books?
- Lets assume that we know the url's or each book, we want to **automate** the **downloading** and **saving** of our books
  - use part of the URL as the file name

```
import urllib.request

onlineBooks = ["http://textfiles.com/etext/FICTION/alicewonder.txt",
"http://textfiles.com/etext/MODERN/hckr_hnd.txt",
"http://textfiles.com/etext/AUTHORS/SHAKESPEARE/shakespeare-macbeth-
46.txt"]

for url in onlineBooks:
        text = str(urllib.request.urlopen(url).read())
        ……
        ……
```

Python I/O – Solution

```
import urllib.request

onlineBooks = ["http://textfiles.com/etext/FICTION/alicewonder.txt",

"http://textfiles.com/etext/MODERN/hckr_hnd.txt",

"http://textfiles.com/etext/AUTHORS/SHAKESPEARE/shakespeare-macbeth-
46.txt"]

for url in onlineBooks:
        text = str(urllib.request.urlopen(url).read())
        locationOfFS = url.rfind("/")
        file = open(url[locationOfFS + 1:], "w")
        file.write(text)
        file.close()
```

Python I/O – Class Example 4

- Lets create a logging system for a company.

- Simple menu that has two options
  - Create log including a date time stamp (appends it to the log file)
  - Show logs

```
import time

menu = 0
while menu != 3:
    print("*****************")
    print("*   MENU        *")
    print("*****************")
    print("* 1) Create Log *")
    print("* 2) View Logs  *")
    print("* 3) Exit       *")
    print("*****************")
    menu = int(input("Please enter option:"))

    if menu == 1:
        log = input("Please enter log:")
        log = time.strftime("%d/%m/2%Y at %H:%M:%S") + "  Log: " + log
```

Python I/O – Solution

```
import time

menu = 0
while menu != 3:
    print("*****************")
    print("*   MENU        *")
    print("*****************")
    print("* 1) Create Log *")
    print("* 2) View Logs  *")
    print("* 3) Exit       *")
    print("*****************")
    menu = int(input("Please enter option:"))

    if menu == 1:
        log = input("Please enter log:")
        log = time.strftime("%d/%m/2%Y at %H:%M:%S") + "  Log: " + log
        file = open("businessLog.txt", "a")
        file.write("\n" + log)
        file.write("\n----------------------------------------\n")
        file.close()

    if menu == 2:
        file = open("businessLog.txt", "r")
        text = file.read()
        print(text)
        file.close()
```

Python I/O – Splitting Data

- The book a tale of two cities, is a very un-edited piece of text.
- For example it contains many issues, for example multiple spaces, \n which are not helpful for analysis.
- Next we would like to count occurrences or words, such as "**the**"?
- How could we do that previously?
- Must use split (this results in a list)

Python I/O – Splitting Data

- In the below example we split the data into a list based on spaces:

```
file = open("aCityOfTwoTails.txt“, “r”)
text = file.read()
file.close()

allWords = text.split(" ")

for word in allWords:
```

- Now that the data is split, into words (its not perfect), try and count the occurrences of the word “**the**”.

Python I/O – Solution

```
file = open("aCityOfTwoTails.txt“, “r”)
text = file.read()
file.close()

allWords = text.split(" ")

numOfOccur = 0
for word in allWords:
    if word.lower() == "the":
        numOfOccur += 1

print(numOfOccur)
```

Python I/O – with

- "***with***" is an an additional way to read/write a file.

- No close required
- Better error handling
- Uses other standard methods such as a for loop.

```
with open("aCityOfTwoTails.txt", "r") as file:
    text = file.read()

print(text)
```

Python I/O – CSV Files

- A .csv file is a staple in data
- Most systems (including databases export to a csv file)
- What is a csv file?

```
item00,item01,item02,item03\n
item10,item11,item12,item13\n
```

Python I/O – CSV Files

- We could split this up manually
- Lets look at the file on Moodle, myCSV.txt

```
with open("myCSV.txt", "r") as file:
    my1DListOfCSV = file.read().split("\n")

print(my2DListOfCSV)
```

- This split gives us each line (really just a 1D list)

Python I/O – CSV Files

- We could split this up manually
- 2D List using the for loop that the ***with*** feature gives:

```
my2DListOfCSV = []

with open("myCSV.txt", "r") as file:
    for line in file:
        my2DListOfCSV.append(line.split(","))

print(my2DListOfCSV)
```

Python I/O – CSV Files

- Using the built in CSV reader:

```
import csv

with open("myCSV.txt", "r") as file:
    my2DListOfCSV = list(csv.reader(file))

print(my2DListOfCSV)
```

Python I/O – CSV Files

- Writing to a csv file (appending in this case):

```
import csv

name = input("Please enter name:")
age = int(input("Please enter age:"))
height = float(input("Please enter height:"))

with open("testCSV.csv", "a", newline=' ') as file:
    singleEntry = [name, age, height]
    writer = csv.writer(file)
    writer.writerow(singleEntry)
```

- The new line option is due to windows always adding an additional "\n"

Python I/O – *In class*

- Using the below code, enter 5 names, ages and heights.

```
import csv

name = input("Please enter name:")
age = int(input("Please enter age:"))
height = float(input("Please enter height:"))

with open("testCSV.csv", "a", newline=' ') as file:
    singleEntry = [name, age, height]
    writer = csv.writer(file)
    writer.writerow(singleEntry)
```

- Then open the file and print the average, minimum and maximum height.
- Do not forget to cast values to int's, as python can not know in advance their types

Python I/O – Solution

```
with open("testCSV.csv", "r") as file:
    people = list(csv.reader(file))

maxAge = int(people[0][1])
minAge = int(people[0][1])
total = 0
for person in people:
    if int(person[1]) > maxAge:
        maxAge = int(person[1])
    if int(person[1]) < minAge:
        minAge = int(person[1])
    total += int(person[1])

print("Max age:     ", maxAge)
print("Min age:     ", minAge)
average = round(total / len(people), 2)
print("Average age:", average)
```